AMENDMENTS TO THE CLAIMS

Please amend the claims to be as follows, where markings are included to show changes made.

1. (currently amended) A method of rapidly selecting a physical memory locality to accomplish efficient memory allocation in a multiprocessor system, the method comprising:

receiving a locality request from a virtual memory fault handler, the locality request including an indication of a search policy to use from among a plurality of search policies;

forming ~~providing~~ a data structure based on physical memory localities within the system and the search policy that was indicated, said data structure including sets of equidistant physical memory localities; and

selecting a preferred physical memory locality using a pointer to a locality within said data structure[[,]].

~~wherein the pointer is rotated amongst localities within a current equidistant set so as to provide for round-robin type selection amongst those equidistant physical memory localities.~~

2. (canceled)

3. (currently amended) The method of ~~claim 2~~ claim 1, wherein the physical memory localities include local memories at cells in the system.

4. (currently amended The method of claim 3, wherein the ~~search policy comprises~~ plurality of search policies includes a "closest first" policy.

5. (original) The method of claim 3, wherein the physical memory localities further includes interleaved memory in the system.

6. (currently amended) The method of claim 5, wherein the ~~search policy comprises~~ plurality of search policies further includes an "interleaved first" type of policy.

7. (previously presented) The method of claim 1, wherein the selection of the preferred locality is performed using a get "best"/"next best" iteration procedure.

8. (currently amended) The method of claim 1, wherein the pointer is rotated amongst localities within a current equidistant set so as to provide for round-robin type selection amongst those equidistant physical memory localities

9. (original) The method of claim 1, wherein the determination of the preferred locality includes changing to a next equidistant set if there is no memory available in any locality of a current equidistant set.

10. (original) The method of claim 9, further comprising returning an indication that no locality is available if no locality within any of the equidistant sets has sufficient memory.

11. (currently amended) A multiprocessor computing system, the system comprising:
multiple symmetric multiprocessing (SMP) nodes;
multiple central processing units (CPUs) at each SMP node;
a memory control unit at each SMP node which is coupled to each CPU at that SMP node;

shared memory at each SMP node which is accessible by way of the memory control unit at that SMP node;

a switching system coupled to the memory control units so as to interconnect the multiple SMP nodes;

an operating system running on the CPUs;

a virtual memory (VM) fault handler within the operating system; and

a VM locality module within the operating system[[; and]],

~~a data structure including sets of equidistant physical memory localities,~~

wherein the VM locality module is configured to receive a locality request from the VM fault handler, the locality request including an indication of a search policy to use from among a plurality of search policies, and is further configured to form a data structure based on the search policy that was indicated. ~~determines a preferred locality using a pointer to a locality within the data structure, and~~

~~wherein the pointer is rotated amongst physical memory localities within a current equidistant set so as to provide for round-robin type selection amongst those equidistant physical memory localities.~~

12. (currently amended) The system of claim 11, wherein the VM locality module is further configured to determine a preferred locality using a pointer to a locality within the data structure.

13. (currently amended) The system of ~~claim 11~~ claim 12, wherein the shared memory includes both local memory and interleaved memory, and wherein the ~~preferred locality is determined using a "closest first" search policy~~ plurality of search policies include at least a closest first search policy and an interleaved first search policy.

14. (currently amended) The system of claim 13, wherein the data structure for the closest first search policy comprises a first set including a closest local memory locality and one or more other sets of equidistant localities.

15. (original) The system of claim 14, wherein the other sets include an interleaved memory locality.

16. (canceled)

17. (canceled)

18. (currently amended) The system of ~~claim 17~~ claim 13, wherein the data structure for the interleaved first search policy comprises a first set including an interleaved memory locality and a set including local memory localities.

19. (canceled)

20. (canceled)

21. (canceled)

22. (currently amended) A multiprocessor computing system configured so as to rapidly select physical memory localities to accomplish efficient memory allocation, the multiprocessor computing system comprising an operating system which includes:

a virtual memory manager configured for extending a memory space beyond limits of a physical address space;

a virtual memory fault handler configured to interrupt execution of the virtual memory manager when a page fault occurs; and

a virtual memory locality module configured to receive a locality request from the virtual memory fault handler, to form a data structure having sets of equidistant physical memory based on a search policy indicated in the locality request, and to rapidly select a physical memory locality in the system using a pointer to [[a]] the data

structure. ~~having sets of equidistant physical memory localities, wherein the pointer is rotated amongst physical memory localities within a current equidistant set so as to provide for round-robin type selection amongst those equidistant physical memory localities; and~~

~~a virtual memory fault handler configured to interrupt execution of the virtual memory manager when a page fault occurs and to utilize the virtual memory locality module to determine the physical memory locality from which to allocate memory in response to the page fault.~~